File No. S360-30
Form Y28-6607-5

Program Logic

IBM System/360 Operating System
Direct Access Device Space Management

Program Number 360S-DM-508

This manual provides detailed information on direct access device space management (DADSM) routines. These routines control the use of external direct access storage by maintaining the information in data set control blocks. The functions and structures of the routines are described, as are their relationships to other portions of IBM System/360 Operating System. This manual also describes the structure of volume tables of contents which are processed by DADSM routines.

It is intended for use by persons involved in program maintenance, and system programmers who are altering the program design. It can be used to locate specific areas of the program, and it enables the reader to relate these areas to the corresponding program listings. Program logic information is not necessary for the use and operation of the program; therefore, distribution of this publication is limited to those with the aforementioned requirements.

Restricted Distribution

Sixth Edition (1969)

This publication corresponds to Release 18. It is a major revision of, and obsoletes, Form Y28-6607-4. Changes to this manual reflect:

- A DOS to OS VTOC conversion routine has been added to the allocate routine. With the addition of this routine, it is possible to use standard OS space allocation techniques (TRK, CYL, or blocksize) on a volume which had previously been used under DOS. Before these routines were added, space could be allocated on such a volume only through the ABSTR (absolute) or SUBALLOC (suballocation) techniques.

  It should be noted that although the VTOC is converted from DOS format to OS format now, there are still some possible incompatibilities between volumes initialized on DOS systems and those initialized on OS systems. OS stand-alone utilities may not be able to process DOS volumes if alternate tracks have been assigned under DOS.

- A new module, IGG0325L, has been added to the allocate routine. This module performs some of the functions previously performed by IGG0325E for split-cylinder allocation requests.

New or modified text in this edition is indicated by a vertical bar in the left-hand margin when the changes are small and by the symbol ● next to the page number when the entire page has been redone.

Changed or added illustrations are indicated by the symbol ● to the left of the caption. Modified sections of the illustrations are also indicated by a bar in the left-hand margin where this has been practical.

Specifications contained herein are subject to change from time to time. Any such change will be reported in subsequent revisions or Technical Newsletters.

This publication was prepared for production using an IBM computer to update the text and to control the page and line format. Page impressions for photo-offset printing were obtained from an IBM 1403 Printer using a special print chain.

A form for readers' comments appears at the back of this publication. It may be mailed directly to IBM. Address any additional comments concerning this publication to the IBM Corporation, Programming Publications, Department D78, San Jose, California 98114.

PREFACE

The information contained in this manual is intended for programmers engaged in maintenance of DADSM routines. Because of the close relationship between DADSM and other portions of the control programs of IBM System/360 Operating System, many details in this manual are helpful in understanding overall system operations.

This manual is arranged in two sections. The first describes the structure of a Volume Table of Contents and how space is accounted for in data set control blocks. The second is the description of the DADSM functions and the internal logic of the routines that perform the functions.

The manual is a detailed and comprehensive guide to the internal structure and functions of the direct access device space management routines. It is designed to be used in conjunction with the assembly listing and, consequently, does not discuss program structure at the machine instruction level.

PREREQUISITE READING

For general background information about the System/360 Operating System:

IBM System/360 Operating System: Concepts and Facilities, Form C28-6535

For information about the DADSM-user interface:

IBM System/360 Operating System: Supervisor and Data Management Services, Form C28-6646

For information about the primary control program:

IBM System/360 Operating System: Introduction to Control Program Logic, Program Logic Manual, Form Y28-6605

For information about the MVT configuration of the control program:

IBM System/360 Operating System: Control Program Logic Summary, Form Y28-6658

This page intentionally left blank.

CONTENTS

FIGURES

This page intentionally left blank.

Direct access device space management (DADSM) consists of routines that allocate space to data sets on direct access volumes. DADSM performs this function by maintaining the volume table of contents (VTOC), a data set that is included in every direct access volume by the volume initialization utility program (IBCDASDI). A VTOC contains a data set control block (DSCB) for each data set on the volume and for all unused space on the volume.

DADSM routines update VTOCs by creating DSCBs for new data sets and deleting the DSCBs of data sets purged from storage. When a data set is created or an existing data set is enlarged, DADSM finds unused space for it by searching the appropriate DSCB in the VTOC, allocating the space as the extent of the data set, and removing it from available space. When DADSM deletes a data set, it also removes the DSCB of the data set from the VTOC; and the extent of the data set identified in the DSCB, is again available for future allocation. DADSM can also return unused space at the end of a data set extent to available space by updating the DSCB of the data set.

DADSM consists of seven routines; the following four perform the space accounting functions:

- The allocate routine allocates initial space to data sets.
- The extend routine allocates additional space to a data set at end-of-volume.
- The scratch routine deletes data sets by deleting their respective data set control blocks.
- The release routine returns unused space to available storage when a data set is closed.

The remaining DADSM routines are: rename, which changes the name of a data set; obtain, which is used to gain direct access to any block in the VTOC; and Lspace, which provides information to the caller about the amount of space available on a volume.

The DADSM routines are all transient SVC routines. Allocate, obtain, scratch, rename, and Lspace can be called directly by an SVC instruction. However, the allocate SVC can only be issued by either the job scheduler or an Operating System/360 utility. The extend routine is entered from a module of the End of Volume SVC routine or the catalog routine by means of an XCTL macro instruction. The release routine is entered (via an XCTL macro instruction) from a module of the close routine. Obtain, scratch and rename are called via system macro instructions.

Appendix C contains a summary of the functions of each routine and the method by which each routine is invoked.

The methods of using the obtain, scratch, and rename routines are described in the publication IBM System/360 Operating System: System Programmer's Guide, Form C28-6550.

Components of the operating system use DADSM routines when space management functions are requested implicitly or explicitly in DD statements. The space requested for a new data set is allocated for job management by the allocate routine. When an old data set is updated and would overflow its boundaries, the end-of-volume routine of I/O support requests new space for the data set by calling the extend routine (only if a secondary quantity was specified in the data set's DD statement and if the data set is not organized in an indexed sequential manner). The CVOL routine of catalog management uses the extend routine to allocate additional space for a catalog data set. If a DD statement specifies that a data set is to be deleted at the end of a job, the scratch routine, called by job management, deletes the data set. The catalog routine uses the scratch routine when an uncataloging operation must delete a data set of a generation data group. The release routine is used by the close routine of I/O support, and the obtain routine is used by job management to find a model DSCB for a new data set of a generation data group. Utility programs use the scratch, allocate, rename, and obtain routines.

DADSM routines transfer control between modules via the XCTL macro instruction. The allocate and scratch routines use the directory entry (DE) form of the macro instruction. This form requires that the IEHIOSUP utility program insert the relative disk address in the modules when the system is generated. All other DADSM routines use the name form of the macro instruction.

Before considering the DADSM routines and their operation, it is necessary to understand the structure of the VTOC and the method of recording the use or availability of direct access storage space in the operating system.

A volume table of contents is a data set that contains a DSCB for every data set and for all available space on a direct access volume. The VTOC, which varies in length from one volume to another, can be located anywhere on a volume following IPL records and volume label(s). Its starting address is recorded in the standard volume label.

The extent of a VTOC must be one area of contiguous tracks on one or more contiguous cylinders.

THE DATA SET CONTROL BLOCK

For each data set on a direct access volume, there must be a corresponding data set control block (DSCB) in the VTOC of that volume. A DSCB, which describes the attributes and extents of a data set, consists of up to three physical blocks chained together to form one logical record (DSCB) in a VTOC. Each block is 140 bytes long (a 44-byte key and a 96-byte data portion), and contains information used by data management to control access to a data set. If a data set resides on more than one volume, there must be a DSCB for the data set in the VTOC of each volume.

DSCBs consist of blocks of which there are seven formats (see the publication IBM System/360 Operating System, System Control Blocks, Form C28-6628, for a more complete description of the DSCB):

- A format 1 block can identify any data set, except for the VTOC, on direct access storage. It can identify up to three noncontiguous areas of a data set. One format 2 and/or one format 3 block can be chained to a format 1 block to constitute one DSCB. There must be a format 1 block for each data set (or part of a data set) on each volume.
- A format 2 block describes an indexed sequential data set. A format 2 block, if used, must be chained to a format 1 block.
- A format 3 block describes a data set of more than three noncontiguous areas. A format 3 block, if used, must be chained to a format 1 or format 2 block.
- A format 4 block describes the VTOC. It always appears first in any VTOC. One format 4 block constitutes one DSCB and can not be chained to other blocks.
- A format 5 block describes up to 26 noncontiguous areas that are available for allocation on a volume. Each area is indicated in a separate "extent entry" in the block. Format 5 blocks can be chained together if the volume contains more than 26 available areas.
- A format 6 block describes up to 26 split-cylinder data set extents. This block has the same format as the format 5 block, but describes extents shared by more than one data set. Each extent is identified in a separate "extent entry" in the block.
- A format 0 block is available space in the VTOC. The block contains all zeros, and can be imagined as a "hole." When a data set is deleted from a volume, a format 0 block is written over the DSCB of the data set.

After initialization, and before data has been placed on a volume, the volume contains one data set: the VTOC. The space allocated to the VTOC is described in the format 4 DSCB at the beginning of the VTOC. The remaining space on the volume is described in the format 5 block that follows the format 4 block. That space is available to be subdivided and allocated to data sets when they are created or extended.

If, after initialization, the VTOC is recorded on cylinder 0 following the volume labels or recorded at the end of the volume, all the available space on the volume is one area of contiguous storage extending from the end of the VTOC to the end of the volume or from the volume labels to the beginning of the VTOC. If the VTOC does not follow immediately after the volume labels or appear at the end of the volume, available space is two areas of contiguous storage; one area in front of and one area following the VTOC. Each area is accounted for in a separate format 5 extent entry. The entry contains the address of the first track of the extent, the number of full cylinders in the extent, and the number of tracks in addition to the full cylinders.

Except on disk operating system volumes and on volumes containing split cylinder data sets, the total number of tracks accounted for in all DSCBs of a VTOC, at any time, is the total number of tracks on the volume. The unused tracks are identified in the format 5 DSCB, and used tracks are identified in blocks of format 1, 3, and 4 of data set DSCBs.

When a data set is created, the allocate routine finds space on the volume by searching the format 5 DSCB. When space is found, the amount needed is subtracted from the space described in the format 5 DSCB. A new DSCB is created for the new data set and is placed in the VTOC in the first available hole (format 0 block). When a data set is deleted, its format 1, format 2, and format 3 blocks are replaced by format 0 blocks (holes), and the extent used by the data set is returned to available space (i.e., added back into the format 5 DSCB).

The DSCB of one data set consists of one, two, or three blocks, depending on the access method used to process the data set, and on the number of noncontiguous areas in the data set. The blocks are chained together in the VTOC in the following sequences:

1. A format 1 block alone for a data set of not more than three noncontiguous areas.

2. A format 3 block chained to a format 1 block for a data set with more than three noncontiguous areas.

3. A format 2 block chained to a format 1 block for an indexed sequential data set of no more than three noncontiguous areas.

4. A format 3 block chained to a format 2 block, which is chained to a format 1 block, for an indexed sequential data set of more than three noncontiguous areas.

The blocks of one DSCB do not necessarily appear in the VTOC as contiguous blocks or in a defined sequence. Except for the first and second DSCB blocks in a VTOC, new blocks are placed in the hole nearest the beginning of the VTOC when they are created. The relationship of two blocks is shown by chain addresses.

VOLUME TABLE OF CONTENTS CAPACITY

When allocating space, DADSM must consider the availability of space for both the data set's extent on the volume and the data set's DSCB in the VTOC. The space required for the expansion of format 5 and format 6 DSCBs must also be considered.

The format 4 DSCB contains a count of the number of holes in the VTOC. Before allocating space, DADSM routines determine if there are enough holes in the VTOC to accommodate the new DSCB. If there are not enough holes, no allocation is permitted. No provision is made for the dynamic extension of a VTOC. However, old data sets can be deleted and their DSCBs removed from the VTOC.

The IBM 2311 Disk Storage volumes allow 150 physical blocks to be written in a VTOC recorded on cylinder 0 following IPL and volume label(s). If the VTOC is recorded beginning on any other cylinder, one full cylinder can contain 160 physical blocks.

The IBM 2301 Drum Storage volumes allow 63 physical blocks to be written in a VTOC consisting of one track.

The IBM 2321 Data Cell Storage volumes allow 160 physical blocks to be written in a VTOC consisting of one full cylinder.

VOLUME TABLE OF CONTENTS INTEGRITY

With a one-CPU (central processing unit) operating system environment, two or more tasks may concurrently require access to the same VTOC for the purpose of reading or updating (i.e., adding, deleting, or modifying DSCBs in) that VTOC. If more than one CPU has access to the same device or devices, it becomes necessary to protect a volume's VTOC on a device not only between intra-system task accesses but between inter-system accesses as well. To preserve the integrity of the VTOC while one task is using it, DADSM routines use the RESERVE, the ENQ, and the DEQ macro instructions. These macro instructions provide exclusive control of the VTOC system resource for the task issuing the macro instruction. Depending on the macro instruction, either the 'set-must-complete' option or the 'release-must-complete' option may be specified in the operand of the macro instruction. The allocate, extend, scratch, rename, and release routines of DADSM use these macro instructions.

Note: When operating in an environment in which direct access storage devices are not shared, the RESERVE macro instruction defaults to (acts as) an ENQ macro instruction.

The allocate routine, the extend routine, the scratch routine, and the release routine are concerned with the accounting of used and unused space on volumes. The routines are responsible for the creation and deletion of all DSCBs, except format 4, in the operating system. (Format 4 DSCBs are created by the utility program that initializes volumes.) This section discusses the internal logic of these routines by explaining their logical operations and the functional relationships of the routines' modules.

THE ALLOCATE ROUTINE

The allocate routine finds unused space on direct access storage volumes and makes that space available for a new data set. It creates a DSCB for the data set and places in it descriptions of direct access storage areas. The descriptions of those areas are removed from format 5 extent entries. The routine performs special operations for split cylinder data sets, for sub-allocation of data sets, and for allocating space for indexed sequential data sets.

Volumes used by the Disk Operating System (DOS) are processed differently from those of the Operating System, since DOS volumes do not contain complete accounting information. Before the allocation routine attempts to allocate space for new data sets on these volumes, it builds the blocks necessary to provide complete space accounting information. This process is discussed in the section "DOS to OS VTOC Conversion."

Although special methods must be used when allocating split cylinder data sets, or when sub-allocating or processing absolute track requests, there is one general method of allocation.

GENERAL ALLOCATION METHOD

When space is needed for a new data set, the allocate routine looks for the first format 5 block in the VTOC of the specified volume. Beginning with that block and continuing to other format 5 blocks through chain addresses, the routine searches for the amount of space needed by the data set. If enough space is found, the format 5 blocks are updated; that is, the identification of the allocated tracks is removed from the blocks. If the volume does not contain enough space for the data set, the allocate routine returns control to the calling routine. If the data set is not an indexed sequential data set, the calling routine can then specify a new volume to be used for the data set.

If an entire contiguous area is allocated to the data set, the format 5 extent entry for that area is completely removed from the format 5 block. The space thus vacated in the block is removed by the moving up of all subsequent format 5 extent entries in that block and other format 5 blocks that may be chained to it. If only part of an available area is allocated, the space is taken from the beginning of the area, and the lower boundary of the available area becomes the track following the allocated area.

If the space to be allocated begins at a specified track address or on a cylinder boundary, the allocation may be made from within an available area. In that case, the allocation must create a new format 5 extent entry, and the spaces preceding and following the allocated space will become two areas identified in two format 5 extent entries. This causes all subsequent format 5 block entries in that and any chained format 5 blocks to be moved back one location.

A contiguous area that is allocated to a data set is accounted for in the DSCB of the new data set. Extent boundaries are placed in a format 1 or format 3 block of the DSCB.

Except when fulfilling either absolute track requests (see the section Absolute Allocation) or sub-allocation requests (see the section The Sub-Allocation Method), a request for a user label causes an additional track to be allocated to the data set. A track used for a user label always appears as the first extent of a format 1 block.

Besides creating a new DSCB for a data set and updating format 5 blocks, the allocate routine updates the format 4 DSCB of the VTOC. The count of the number of holes in the VTOC must be updated after each allocation. The address of the last format 1 block may have to be changed if the allocate routine placed a new format 1 block at a higher address. In this publication, the address of the last format 1 block is referred to as the "high water mark" of the VTOC. The address is used to terminate channel programs processing the VTOC.

Figure 1 shows the methods of data set space allocation used for the various options specified in the DD statement.

Option	Allocation Method
MXIG	Allocate all the storage space in the single largest area on the volume, if that area is equal to or larger than request.
CONTIG	Allocate the exact amount of space requested from one area on the volume that is equal to or larger than the request.
ALX	Build a list of up to 5 areas of available storage that are the largest on the volume. Allocate all the space in as many of the areas found that are equal to or larger than the request.
No Option Specified (blank)	Build a list of up to 5 areas that are smaller than but closest in size to the request. If, while building the list, there is found an area equal to the request, allocate that area and disregard the list. If no areas are found equal to the request but areas are found larger than the request, allocate the amount of space requested from the area that is larger than but closest in size to the request, and disregard the list. If all available areas are smaller than the request, and if enough space is available in the areas in the list, allocate the exact amount of the request from among the largest areas found.

Note: For an indexed sequential data set only the CONTIG and the (blank) options are valid.

●Figure 1. Initial Allocation Methods

THE SUB-ALLOCATION METHOD

The allocate routine must perform special processing to sub-allocate space for a new data set: instead of removing available space from a format 5 DSCB, the routine removes space from a data set DSCB and allocates it to the new data set. No options can be specified for sub-allocation. The space removed is from the beginning of the data set. The data set so subdivided must have an extent of only one contiguous area, and the data set created by the subdivision can not contain more than one area unless a user label has been requested. In this case, the first track of the sub-allocated space appears as the first extent of the format 1 block.

Sub-allocation is not permitted for indexed sequential data sets.

SPLIT CYLINDER ALLOCATION

Allocation of split cylinder space is handled differently from allocations of track, record, or cylinder requests: The space is allocated as one extent on contiguous full cylinders. The entire extent is removed from available space and accounted for in one entry in the format 6 DSCB. The extent is divided among the split-cylinder data sets, and the part used by each individual data set is accounted for in the DSCB of each data set. The entire extent remains unavailable for future allocation until all the split cylinder data sets have been deleted. The entry in the format 6 DSCB is then removed and the extent is added back into the format 5 DSCB.

The space allocated to split-cylinder data sets is recorded in one entry in a format 6 DSCB, as well as in the DSCBs of the data sets. The space is allocated in full cylinders, and only the original data sets can reside on those cylinders (even if the split-cylinder data sets do not occupy 100% of the cylinders). The first four bytes of the 5-byte extent entry in a format 6 DSCB contain the starting cylinder address and the number of full cylinders in the extent. The first two bytes contain the starting cylinder address in the form of RTA (as opposed to CC, as in split cylinder), and the second two bytes indicate the number of full cylinders being shared. The fifth byte lists the number of data sets sharing the extent. The DSCB of each data set contains an entry that identifies the first and last cylinders of the extent and the address of the first and last track that the data set uses on every cylinder.

If a user label has been requested along with a split-cylinder allocation request, an additional track is allocated for each data set that requires a user label. This additional track constitutes an extent and appears as the first extent of a format 1 block. The space occupied by the user label is not recorded in the format 6 DSCB.

If a request for split-cylinder allocation has been made for a case in which drum storage devices are being used, the allocation is made in the form of a series of contiguous tracks. The tracks allocated to a given data set are contiguous, and the data sets themselves are contiguous within the allocated space. For requests of this type, format 6 DSCBs are not considered since the cylinder 'concept' is not used for drum devices.

THE ALLOCATE ROUTINE MODULES

The allocate routine modules are loaded separately into the transient area of main storage and operate in supervisor mode. Figure 2 shows the modules of the allocate routine and the flow of control through the modules. Input to the routine consists of either a DSCB or a job file control block (JFCB) for the new data set. If the input is a DSCB, the allocate routine is requested to create a data set with a similar DSCB. The input also lists the type of request (records, tracks, or cylinders), and the serial number of the volume on which space is to be allocated. The input is available to the allocate routine through general registers that contain the addresses of pertinent information.

Control passes through the modules of the allocate routine in different paths, depending on the function to be performed. Data sets that are not indexed sequential data sets have space allocated through the use of a certain group of allocate modules. For an indexed sequential data set, another group of allocate modules is used for space allocation. This second group of modules is referred to as ISAM (indexed-sequential access method) allocate modules. In both groups of modules, the first module, IGC0003B, is the same.

Figure 3 lists the sequences in which modules are executed for various functions. Modules of the allocate routine for data sets that are not indexed sequential data sets use the XCTL table in the transferring of control from one module to another. Appendix B contains a description of the XCTL table.

<u>Allocation for Non Indexed Sequential Data Sets</u>

Allocation of each non indexed sequential data set is accomplished in one pass through the allocate routine. The job scheduler issues one SVC 32 for each data set for which space must be allocated. Module IGC0003B is always the first module of the allocate routine to get control.

<u>DUPLICATE NAME SEARCH (MODULE IGC0003B):</u> Entry to the allocate routine is through module IGC0003B. That module performs housekeeping, gets a work area in a protected subpool, issues a RESERVE macro instruction for the VTOC, and initiates a channel program. The channel program reads the format 4 DSCB and the first format 5 block from the VTOC of the first specified volume and searches the VTOC for a DSCB with the same name as the data set being created. The search operation is called the duplicate name search. If a duplicate name is not found, the allocate routine can perform initial allocation of space for the new data set. If a duplicate name is found, initial allocation has already been performed; an error condition is indicated and a message is given.

When a Disk Operating System volume is introduced into the Operating System, control is passed to module IGG0325Z to begin DOS to OS conversion. The DOS to OS VTOC Conversion routine returns control to IGC0003B after the VTOC has been converted. If conversion is required, it takes place before the duplicate name search.

Control is passed from module IGC0003B according to the conditions given in Figure 4. In this figure, the conditions are determined in a left-to-right order.

<u>REQUEST CONVERSION (MODULE IGG0325B):</u> All requests made to the allocate routine are allocated in tracks. Record and cylinder requests are converted to track requests, cylinder requests are aligned on cylinder boundaries, and record requests are aligned on cylinder boundaries if the "ROUND" option is specified.

Request conversion is performed in module IGG0325B.

<u>USER LABEL ALLOCATION (MODULE IGG0325K):</u> For each data set that requires user label allocation, module IGG0325K allocates one track. Space for this track is obtained from the smallest extent of the first format 5 DSCB. The format 5 DSCB is then updated so that the tracks assigned to user labels are unavailable for allocation as space for satisfying other requests.

<u>TRACK AND CYLINDER ALLOCATION (MODULE IGG0325D):</u> Module IGG0325D allocates tracks and cylinders. The IGG0325D module includes allocation for split-cylinder data sets.

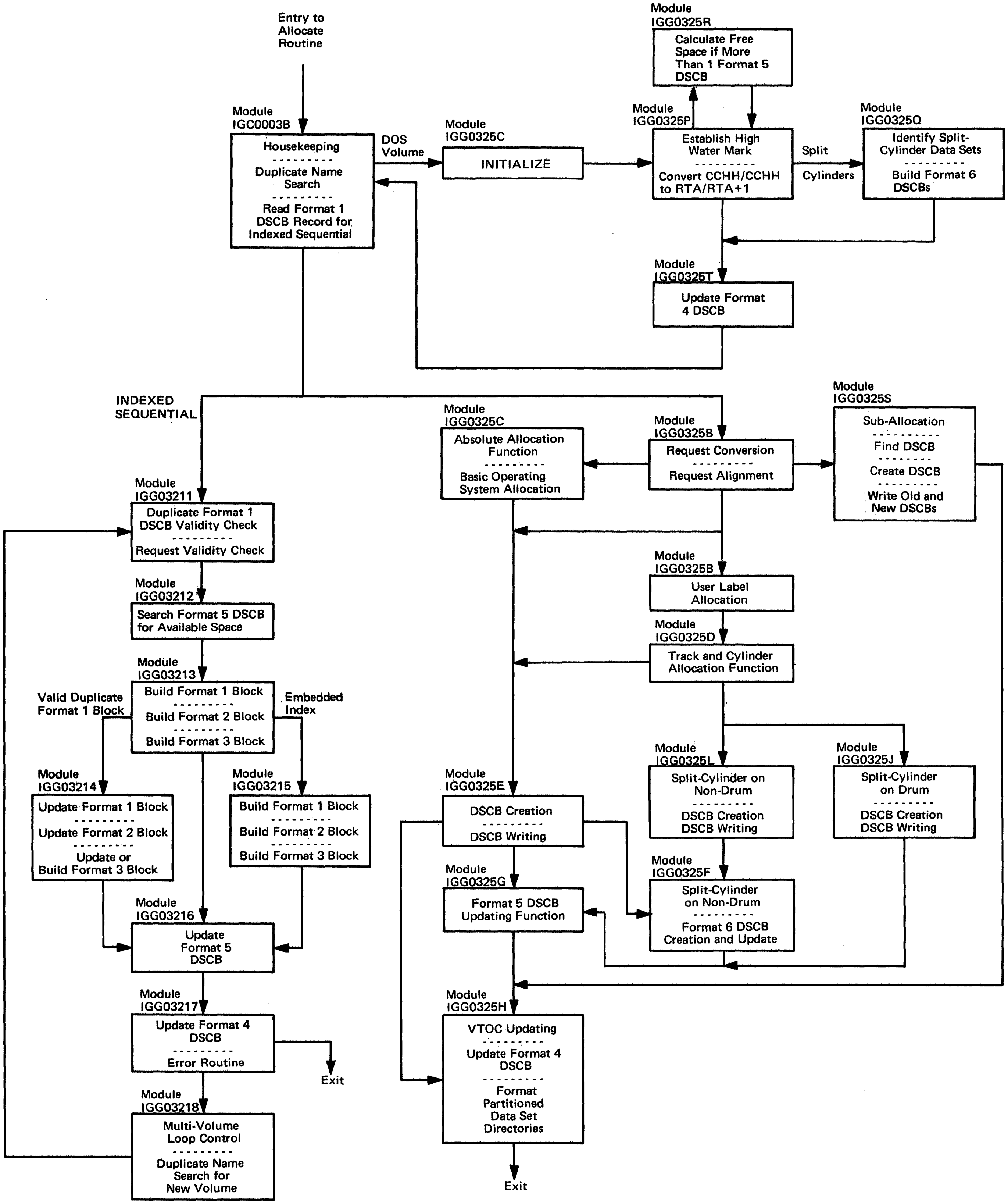

Figure 2. The Allocate Routine Modules

Model DSCB and Normal Track/Cylinder Allocation	Zero Quantity Extent Allocation	Normal Absolute Allocation	Split-Cylinder Allocation (Non-Drum)	Split-Cylinder Allocation (Drum)	Sub-Allocation
IGC0003B	IGC0003B	IGC0003B	IGC0003B	IGC0003B	IGC0003B
IGG0325B	IGG0325B	IGG0325B	IGG0325B	IGG0325B	IGG0325B
→IGG0325D	→IGG0325E	→IGG0325C	→IGG0325D	→IGG0325D	→IGG0325S
IGG0325E	IGG0325H	IGG0325E	IGG0325L	IGG0325J	IGG0325H
IGG0325G		IGG0325G	IGG0325F	IGG0325G	(CVOL Routine)
IGG0325H		IGG0325H	IGG0325G	IGG0325H	
(CVOL Routine)		(CVOL Routine)	IGG0325H		

ISAM Allocation With No Embedded Index and No Duplicate Format 1 Block	ISAM Allocation With Embedded Index and No Duplicate Format 1 Block	ISAM Allocation With Duplicate Format 1 Block
IGC0003B	IGC0003B	IGC0003B
IGG032I1	IGG032I1	IGG032I1
IGG032I2	IGG032I2	IGG032I2
IGG032I3	IGG032I3	IGG032I3
IGG032I6	IGG032I5	IGG032I4
IGG032I7	IGG032I6	IGG032I6
IGG032I8	IGG032I7	IGG032I7
		IGG032I8

Note: If user label allocation is requested, module IGG0325K is to be inserted after module IGG0325B as indicated by the arrows.

●Figure 3. The Allocate Routine Module Flow

Has Duplicate Name Been Found	Is Data Set an ISAM Data Set	Control Given To
Yes	Yes	Module IGG032I1
Yes	No	Calling Routine (Error Exit)
No	Yes	Module IGG032I1
No	No	Module IGG0325B

Figure 4. Exits from Module IGC0003B

To determine if enough space exists to fill a request, this module reads in all the format 5 blocks except the first, one at a time. The first format 5 block will be in the work area when this module is entered. (Refer to "Duplicate Name Search (Module IGC0003B).") All entries in format 5 blocks are in XXYYZ format; XX indicates the first track of an available area, YY indicates the number of full cylinders in the area, and Z indicates the number of tracks in the area, in addition to the full cylinders. When a format 5 block is read, all the addresses in the block are converted from XXYYZ format to RTA1/RTA2 format, which is used internally by DADSM. RTA1 is the first relative track address of the area, and RTA2 is the last relative track address of the area, plus one. The RTA1/RTA2 format facilitates the allocation operation, because a converted request is in track format.

Module IGG0325D also builds an extent table, which consists of identifications of the areas to be allocated to the new data set. (The extent table format is described in Appendix A.) These identifications are extracted from the format 5 DSCB. If there is more than one format 5 block, the others are read in, one at a time, and the identifications extracted are placed in the extent table. The table is passed later to subsequent modules. The format 5 blocks are not changed during the building of the table. A later module removes the identification of the areas from the blocks.

Module IGG0325D allocates only full cylinders for split-cylinder data sets and places the identification of those cylinders in the extent table. Later modules will divide the cylinders as requested and update the VTOC.

ABSOLUTE ALLOCATION (MODULE IGG0325C): Absolute allocation is performed by module IGG0325C. Absolute allocation is similar to track and cylinder allocation in that both functions read format 5 blocks and build an extent table of areas to be allocated.

Absolute allocation always begins at the address specified in the input parameters. The space requested is one area of contiguous storage. However, if a user label track is requested, the first extent (for the data set) in the format 1 block consists of the first track of the absolute request and contains the user label. The processing of any format 5 blocks is simply a verification of the availability of the requested extent.

DSCB CREATION (MODULES IGG0325E, IGG0325F, IGG0325L, AND IGG0325J):

The extent table created in either module IGG0325C or IGG0325D is passed to one of three modules:

- IGG0325L if the request is for split-cylinder allocation on a device other than a drum,
- IGG0325J if the request is for split-cylinder allocation on a drum device, and
- IGG0325E in all other cases.

Each of these modules creates and writes format 1 and format 3 DSCBs. They all issue an ENQ macro instruction with the set-must-complete option to protect the VTOC during subsequent updating.

Module IGG0325F gets control after module IGG0325L, that is, after format 1 and format 3 DSCBs have been written for split-cylinder requests on non-drum devices. This module updates the format 6 DSCB by adding the extent entry identifying the split-cylinder extent. It then passes control to module IGG0325G.

If split-cylinder requests are applied to a drum storage device, module IGG0325J receives control from module IGG0325D, thereby bypassing the modules required to honor normal split-cylinder requests. The DSCBs for these requests are constructed by module IGG0325J. A format 1 DSCB is constructed for each data set for which allocation is requested. Each data set is allocated the same total number of tracks as it would have received if the allocation had been made on split cylinders utilizing a device other than a drum. However, the allocation is in the form of contiguous tracks in the same manner as a contiguous-track request. After the DSCBs have been constructed, they are written in the VTOC and control is passed to module IGG0325G.

UPDATING A FORMAT 5 DSCB (MODULE IGG0325G): Module IGG0325G receives control and an extent table from either module IGG0325E or module IGG0325F. Each storage area listed in the extent table has been allocated to a data set, and the identification of each area must be removed from the format 5 DSCB of the VTOC. Module IGG0325G deletes from the format 5 DSCB the area identifications in the extent table.

After each block of a format 5 DSCB is read, the extent entries in the blocks are converted from the XXYYZ form to the RTA1/RTA2 form. Before the blocks are written back into the VTOC, the extent entries are converted back to the XXYYZ form.

UPDATING THE VTOC (MODULE IGG0325H): Module IGG0325H receives control from module IGG0325E (zero quantity), IGG0325G (normal), or IGG0325S (suballocation) and writes an updated format 4 DSCB in the VTOC. The DSCB contains a count of the number of holes in the VTOC and indicates the high water mark of the VTOC. If, as a result of DSCB creation and format 5 updating, the number of new blocks written differs from the number deleted, the hole count in the format 4 DSCB changes. After the format 5 and format 4 DSCBs are updated, module IGG0325H releases the VTOC by issuing a DEQ macro instruction that specifies the release-must-complete option. If the data set was not partitioned, control passes directly to the calling routine. Otherwise, control passes to the CVOL routine of catalog management to write format blocks in the partitioned data set directory.

SUB-ALLOCATION (MODULE IGG0325S): Module IGG0325S, which performs sub-allocation, receives control from module IGG0325B after request conversion has been performed. Module IGG0325S first issues an ENQ macro instruction with the set-must-complete option for the VTOC. Then the module locates the format 1 block from which space is to be extracted and allocates the space from the beginning of the data set's extent. The area boundaries are placed in a new format 1 block and both blocks are written into the VTOC. Control is passed to module IGG0325H.

The new format 1 block created by sub-allocation will contain the area allocated to it as well as any pertinent information from the JFCB of the data set. Note that, unless a user label track has been requested, neither the new nor the old format 1 block used in sub-allocation contains more than one area. If a user label request has been specified for a data set, the format 1 block contains an additional (first) extent that is used to identify the track containing the user label. This track is the first track assigned to the data set during allocation. No information, except extent boundaries, is transferred from the old DSCB to the new format 1 block.

DOS to OS VTOC Conversion

The text below describes each of the DOS to OS VTOC conversion routine modules. Figure 5 illustrates the flow of control from module to module.

IGG0325Z: This is an initialization module. At entry, it sets up the channel programs and zeros out any existing format 5 and format 6 DSCBs. A new format 5 DSCB is created and initialized to reflect the free space that had been available when the volume was initialized and before any data sets were allocated. The highest allocatable track on the volume is calculated from the Device Characteristics Table. The space for volume labels, IPL records, and the VTOC is subtracted from the new format 5 DSCB and control is passed to IGG0325P.

IGG0325P: This module reads sequentially through the VTOC and establishes a high water mark. It converts the extents in the format 1 and format 3 DSCBs from CCHH/CCHH to RTA/RTA+1 form. It converts five extents at a time and places them in the DADSM table. The converted extents are then subtracted from the free space. If there is only one format 5 DSCB, IGG0325P retains control to subtract the extents. If there is more than one format 5 DSCB, control is passed to IGG0325R to subtract the extents. After all extents have been subtracted, control is passed to IGG0325Q if there are split-cylinder data sets on the volume and to IGG0325T if there are no split-cylinder data sets on the volume.

IGG0325R: This module subtracts allocated extents from free space when there is more than one format 5 DSCB on the volume. It always returns control to IGG0325P.

IGG0325Q: This module makes a second pass of the VTOC to the high water mark. It identifies split-cylinder extents and builds the necessary format 6 DSCBs. It then writes the final XXYYZ form of the format 6 DSCBs in the VTOC.

IGG0325T: This module converts the format 5 DSCBs to the final XXYYZ form and writes them in the VTOC. It then sets bits in the format 4 DSCB to indicate that this volume may now be treated by DADSM as a standard OS volume. Finally, it writes the updated format 4 DSCB and returns control to the beginning of IGC0003B where processing continues as if the volume had been an OS volume in the first place.

Allocation for Indexed Sequential (ISAM) Data Sets

There are three types of space requests possible for an indexed sequential data set. One request may be for space for an index area. Another request may be for space for a prime area to contain the main data. A third request type may be for an overflow area to contain data either displaced from or added to the prime area. From one to three requests may be given for a particular data set, but the order in which these requests may be specified is

If one F5 on volume	If more than one F5 on volume	If one F5 and one F6 on volume	If more than one F5 and F6 on volume
IGG0325Z	IGG0325Z	IGG0325Z	IGG0325Z
IGG0325P	IGG0325P	IGG0325P	IGG0325P
IGG0325T	IGG0325R*	IGG0325Q	IGG0325R*
	IGG0325P*	IGG0325T	IGG0325P*
	IGG0325T		IGG0325Q
			IGG0325T

* May go from IGG0325R to IGG0325P more than once.

●Figure 5. The DOS to OS Conversion Routine Module Flow fixed. The requests must be in the order, index, prime, and overflow, if two or more of the three types are used. In all cases, there must be a prime request.

Each space request is entered into the system as part of a data definition (DD) statement. If there are two or more types of requests for a given data set, they must be given in consecutive DD statements, but only the first request may have a DD name. An entry for each DD statement in a job step is placed in a task input/output table (TIOT) created for the step by the job scheduler. These entries occur in the same sequence as they are in the input stream of the job step.

For each allocation request for an ISAM data set, the job scheduler issues a SVC 32 for an allocate routine. When the request has been processed, the allocate routine returns control to the job scheduler.

Since the ISAM allocate routine may be entered by more than one space request for a given ISAM data set, there may be more than one pass through the routine in order to complete the total space request for that data set. If this condition exists, information such as the type of allocation (e.g., track or cylinder) and the type of request (e.g., index, prime, or overflow) must be communicated among the multiple passes. The ISAM allocate routine provides this information by setting indicators in the last six bytes of the DD name field (TIOEDDNM) of the TIOT entries for the second, and, if necessary, the third requests. The TIOT entries for multiple requests are examined during each pass to validate the request parameters. (Refer to "Validity Checking.") Since multiple requests on a single volume are possible, the second or third pass may encounter a duplicate name DSCB that belongs to the current data set and was written out during a previous pass. To distinguish between a duplicate name DSCB of this type and a 'true' duplicate name DSCB representing another data set, the ISAM allocate routine examines the unit control block (UCB) and communication areas of the TIOT entries.

A request may also be made to distribute an ISAM data set on more than one volume. When this happens, an internal multi-volume request loop is set up within the ISAM allocate routine. (See Figure 2.) Control is not given back to the job scheduler until the necessary iterations of the loop have been completed.

The index area can exist on only one volume. The overflow area can exist on only one volume. The prime area may exist on one or more volumes. Since data in the overflow area is in the same format as that in the prime area, both the overflow and the prime areas must be on the same device type, but they may or may not be on the same volume. The index area may or may not be on the same device type as the prime area, and it may or may not be on the same volume as the prime and/or overflow areas.

If the prime area is specified to exist on more than one volume, both the space that is to be allocated to each volume and the number of volumes requested are given in a DD statement. The total space requested can then be computed.

If a user wishes to have an index to a data set embedded (to reduce access arm movement) in the physical center of the data set, this may be specified if the prime area is to reside on only one volume. In this type of request, there will be no separate specification of an index area; rather, the index area will appear as a parameter in the request for a prime area.

Space for indexed sequential data sets is allocated only in terms of an integral number of cylinders. If a request is given for absolute track allocation, the request will be converted to the cylinder equivalent. An error results if the conversion does not result in an integral number of cylinders.

If the type of request specification parameter is not given (i.e., left blank), a prime area will be allocated. If only a prime area is requested, an index and an overflow area also will be generated within the area specified and allocated for the prime request. If both an index area and a prime area are specified, an overflow area also will be generated within the area requested for the prime area. These generations are made by the indexed sequential access method routines. (Refer to the publication IBM System/360 Operating System: Indexed Sequential Access Methods, Program Logic Manual, Form Y28-6618.)

There are nine modules that handle all space allocation for indexed sequential data sets.

DUPLICATE NAME SEARCH (MODULE IGC0003B): The duplicate name search procedure for an indexed sequential data set space allocation request is the same as that for any type of data set allocation. Therefore, the previous description of module IGC0003B is applicable. (Refer to "Non-Indexed Sequential Data Sets.") When the space allocation request is for an indexed sequential data set, control is given to module IGG032I1.

VALIDITY CHECKING (MODULE IGG032I1): Module IGG032I1 may be entered either from the first load module of the allocate routine or from the seventh load module (IGG032I7) of the ISAM allocate routine. Entry from module IGG032I7 is made only if the request is for the allocation of space on more than one volume.

If a duplicate name DSCB is indicated, the validity of this duplicate name DSCB must be checked. A duplicate name DSCB for this data set will be valid if this request is either the second or the third of a sequence of two or three requests for ISAM space allocation and if the same unit control block (UCB) is being used.

A 'true' duplicate name results in a transfer of control to the seventh load module, IGG032I7, of the ISAM allocation routine. (Module IGG032I7 is entered when any error in an ISAM allocation is encountered.)

In checking for invalid requests, module IGG032I1 indicates an error if the first request of a multi-request occurs without a DD name, or if subsequent requests of a multi-request do have a DD name. This information is obtained from the TIOT that is already in storage when the allocate routine is given control.

Other checks performed by this module involve:

1. Determining if the request has been allocated as either an integral number of cylinders or a number of absolute tracks equivalent to an integral number of cylinders.

2. Determining either if the type of space allocation has been specified as contiguous or if no (i.e., 'blank') space allocation specification has been given.

3. Determining if the same form (e.g., cylinders) of space allocation specification has been given on all requests for the data set involved.

4. Determining if the same device type has been specified for both a prime request and an overflow request.

5. Determining if there have been more than one of a given type of request specified. (For example, two requests for a prime area is invalid.)

6. Determining if both an index request and a prime request with an index parameter (i.e., an embedded index request) have been specified for the same data set. If this happens, an error exit is made.

If the current passage through the ISAM allocate routine is the first for a given data set, the need for building a format 2 block is indicated. During subsequent passes through the ISAM allocate routine for the same data set, a format 2 block is not built.

In each of the above checks, an indication of an invalid condition results in a transfer of control to the seventh load module of the ISAM allocate routine.

To determine if there is enough room (i.e., there may be a need for up to four holes to be used by the ISAM allocate function: for format 1, 2, 3, and 5 blocks) for additional blocks, a check is made of the format 4 DSCB. (The format 4 DSCB was read into storage by either module IGC0003B or module IGG032I7, depending on which module gave control to module IGG032I1.) If there is not enough room, an error exit is made.

If a request for a prime area with an embedded index is valid, module IGG032I1 computes the total quantity of space requested, and control is given to module IGG032I2. An invalid embedded index request causes an exit to the seventh load module of the ISAM allocate routine. If no embedded index has been specified and there has been no error in specifying the number of cylinders or tracks for area requests, control is given to the second load module of ISAM allocate.

FORMAT 5 SEARCH (MODULE IGG032I2): The second load module of the ISAM allocate routine searches all available format 5 blocks on a given volume to determine if there is space available to satisfy the given request.

There are three space allocation options that are considered in this module.

- The request may be for a space that has an absolute track starting point and that begins and ends on a cylinder boundary.

- The request may be for contiguous cylinders.

- The request may be for non-contiguous cylinders.

In searching for available space, this module makes a systematic search of the first format 5 block (which has been read into main storage by either the duplicate name search module or the seventh load of the ISAM allocate routine as described in the section "Completion of Processing"). If space is not available in this first block, another format 5 block is read in if one is available. This process is repeated to search all available format 5 blocks on the given volume if necessary. Module IGG032I2 initiates its own channel program to read in all but the first format 5 block. If space cannot be found on the volume, an error indication is set, and this module gives control to the seventh load of ISAM allocate.

The searching process involves comparing sequentially each extent entry in a format 5 DSCB to the parameter requested (i.e., to either a quantity of cylinders, or a given absolute track location plus a quantity of cylinders). As previously indicated, the format 5 DSCB entries are in the XXYYZ form. (Refer to the section "Track and Cylinder Allocation.") However, for the ISAM allocate routine, the entries are not converted to the RTA1/RTA2 form when the format 5 DSCB is read in to storage during the second load module.

For a request having an absolute track starting point, the search routine examines the XX and YY fields of the format 5 DSCB to determine whether the request can be satisfied.

Contiguous cylinder requests are treated in the following manner. For a given volume, the blocks of the format 5 DSCB are searched for an extent that is equal in size to the space requested. As this is done, extents larger in size than the requested space may be encountered. The identification of the first such extent is put in a 'held' area of an assigned work area. As each subsequent larger extent is encountered, its size is compared to the size of the extent being held. The result of the comparison is always to place the identification of the smallest extent (of all these larger extents) in the held area.

If, during the search of the format 5 blocks, an extent exactly equal in size to the space requested is found, that extent is allocated as the requested space and the search ends. Otherwise, the search terminates after all format 5 blocks have been searched. If a larger extent has not been found, an error exit is made. If there is an extent being held, the requested space is allocated from this 'held' quantity. This method improves the allocation process in that the requested space is taken from the smallest available extent that will fulfill the request, thus leaving larger extents intact for other possible allocation requests.

Requests specifying that the allocated cylinders need not be contiguous are handled in the following manner. As the search for space progresses, a five-entry push-down list is built and maintained. This list contains the largest extents, encountered in the search, that are not large enough to contain the complete space request. If an extent larger than the requested space is encountered, the pushdown list is abandoned and a 'held' area is established and maintained as described for the contiguous cylinder requests.

If an extent exactly equal in size to the requested space is encountered, the search is ended and that extent is allocated to the request. If an equal extent is not found, the search terminates after all format 5 blocks have been searched.

The requested space is then allocated either from the 'held' extent as in the case of a contiguous cylinder request or from entries in the push-down list. If the push-down list is used, from two to five of the largest entries in the list are used as needed.

If the request cannot be fulfilled by any of the preceding possibilities, an error exit is made.

When module IGG032I2 determines that an extent table (subsequently referred to as the DADSM table) can be built to contain the format 5 DSCB entries that satisfy the request, the appropriate entries are converted from the XXYYZ form to the RTA1/RTA2 form and placed in the DADSM table. The extent entries in the DADSM table are ordered according to ascending relative track addresses (RTAs).

The form of the format 5 DSCB entries themselves are not changed. A later module in the ISAM allocate routine updates the format 5 DSCB to reflect the removal of the entries from available space. As in allocation routines for other than indexed sequential data sets, the DADSM table is passed to subsequent modules.

After building and sorting the DADSM table, module IGG032I2 gives control to module IGG032I3.

BUILDING DSCBS (MODULE IGG032I3): The general overall functions of this module and of modules IGG032I4 and IGG032I5 are similar. They basically accomplish the building of the DSCBs to represent the space that is allocated to a data set. In an error-free situation, module IGG032I3 will always gain control from module IGG032I2. If there is neither a duplicate name DSCB in storage nor a specification for an embedded index, control remains in the third module of the ISAM allocate routine, and the module issues an ENQ macro instruction with the set-must-complete option for the VTOC.

Upon entry to module IGG032I3, both the DADSM table (constructed by module IGG032I2) and the job file control block (JFCB) are in main storage. If control remains in module IGG032I3 (rather than being passed to either module IGG032I4 or module IGG032I5), the module must build a format 1 block for the data set. The JFCB contains information, pertaining to data set characteristics, that is used by this module in the construction of a format 1 block. The extent (or space identification) information is obtained from the DADSM table. Since the entries in the DADSM table have been entered there in a relative track address (RTA) format, they must be converted to a cylinder (CC) and head (HH) format (given as CCHH) for use as seek addresses by the indexed sequential access method routines. The third module of the ISAM allocate routine gives control to a resident system routine to make the track address conversion that is required. (Refer to the publication IBM System/360 Operating System: Sequential Access Methods, Program Logic Manual, Form Y28-6604.)

The maximum number of extent entries that can be placed in the format 1 block is three. If the DADSM table contains more than three entries, a format 3 block is then constructed. The extents in the format 3 block are constructed in the same manner as for the format 1 block. (See the publication IBM System/360 Operating System: System Control Blocks, Form C28-6628, for a description of the DSCB blocks.)

If a format 3 block has been constructed, a channel program is initiated to search the VTOC for the first available hole and to write out the block. The format 3 block must be constructed and written out before either a format 1 or a format 2 block is written. This is because the address of the format 3 block must be placed in either the format 1 or the format 2 block as a chaining address.

After either the format 1 block has been constructed for the case when the DADSM table contains under four entries or the format 3 block has been written out for the case when the DADSM table contains more than three entries, the need for a format 2 block is determined. A format 2 block contains information pertaining to indexing of the data set, and one will always be constructed at some time during the total allocation of space for the data set involved. The format 2 block for a data set will reside on the same volume as the index area for the data set.

In determining the need for constructing a format 2 block at this point, there are several possibilities to consider. Figure 6 summarizes the factors involved in making the proper decision. Each of the possibilities in Figure 6 is related to the current pass through the ISAM allocate routine.

If a format 2 block is necessary, it is constructed and written out. The ISAM allocate routine only fills in the block identifier fields, the DS2CYLOV field, the DS2NOTRK field, and, if there is a format 3 block, the DS2PTRDS field in the format 2 block. The rest of the block is completed by the indexed sequential access method.

After the format 2 block is written out or if there were no need for a format 2 block in this pass, the format 1 block is written out. If a format 2 block was constructed on this pass, the address of the format 2 block is placed in the format 1 block chaining field. If a format 3 block but no format 2 block has been constructed, the address of the format 3 block is placed in the chaining field of the format 1 block. If neither a format 2 block nor a format 3 block has been constructed, the chaining field is not filled in. Module IGG032I3 then relocates a channel program for reading in a format 5 DSCB and gives control to module IGG032I6.

DUPLICATE FORMAT 1 ACTION (MODULE IGG032I4): When module IGG032I4 receives control from module IGG032I3, there exists in storage both a format 1 block from a previous pass of the ISAM allocate routine and a DADSM table of at least one entry. In addition, there may also be a format 2 block on the same volume. There may also exist a format 3 block if, when module IGG032I4 gets control, a previous pass through the ISAM allocate routine (to satisfy a request associated with this data set) resulted in the construction of more than three entries in a DADSM table.

The basic functions of this module are similar to those of module IGG032I3. An ENQ macro instruction with the set-must-complete option is issued for the VTOC. The existing format 1 block is updated if possible, a format 3 block is either constructed if necessary or is updated if one already exists, and a format 2 block is updated (i.e., pointer field is changed) if necessary.

If the existing format 1 block contains less than three entries, the entries in the current DADSM table that was constructed by module IGG032I2 are placed in the format 1 block until the block is full. When the format 1 block is filled and there exist more entries in the DADSM table, a format 3 block is constructed to contain the overflow entries. As entries are placed in the format 1 block and in the format 3 block, they are converted from the RTA form (of the DADSM table) to the CCHH form. (Refer to "Building DSCBs.")

If the existing format 1 block contains three entries and a format 3 block does not yet exist on the volume, this module constructs a format 3 block and places in it the entries from the DADSM table (again giving control to a system resident conversion routine to change the form of the DADSM table entries).

If the current pass through the ISAM allocate routine has constructed a format 3 block and if a format 2 block already exists on the current volume, a chaining address pointing to the format 3 block is placed in the format 2 block.

If a format 3 block already exists when module IGG032I4 gets control, the entries in the current DADSM table are converted and added to the format 3 block. Gaining access to the format 3 block may involve the format 2 block for the data set if the format 2 and format 3 blocks are on the same volume. When the DADSM table entries have been placed in the format 3 block, the format 3 block is written out in the VTOC. Since the format 2 block already exists and contains a pointer to the format 3 block, the format 2 block does not have to be updated in this situation.

The last block to be written is the format 1 block. The writing of the format 1 block is delayed since a pointer to a format 3 block may have to be inserted.

The channel program that has been used for reading in and writing out of the blocks is relocated and initialized for use by module IGG032I6, and control is then given to module IGG032I6.

EMBEDDED INDEX (MODULE IGG032I5): If a request for prime space with an index area embedded in the middle of the prime area is encountered, module IGG032I5 is used. All of a format 1 block except the up-to-three possible extents is constructed by module IGG032I3. Module IGG032I3 then transfers control to module IGG032I5. In addition to the partially completed format 1 block, a DADSM table with at least one entry is passed to module IGG032I5. This module issues an ENQ macro instruction with the set-must-complete option for the VTOC. Then the module will complete the format 1 block, build a format 3 block if necessary, and build a format 2 block.

This module considers both the total space that is to be allocated (as specified in the space request) and the space that is available to satisfy the request. This latter space has been found on a format 5 DSCB and placed in the DADSM table by the second load of the ISAM allocate routine.

The space requested for the prime area is divided into two parts as nearly equal as cylinder boundary requirements permit. The entries in the DADSM table are then taken in order and assigned as prime extents as necessary to provide allocation for the first half of the prime area. The requested index area is then assigned from the next available track space indicated in the DADSM table. Finally, the remaining space in the DADSM table is assigned to the other half of the prime area request. Depending on the requirements of the request and the sizes of the DADSM table entries, it is possible to form up to seven DSCB extents from the entries in the DADSM table.

Total Space Request for the Data Set (Given in terms of DD card parameters)	Current Request (Given in terms of DD card parameters)	Format 2 Block Needs to be Built	Comments
INDEX, PRIME, OVERFLOW	INDEX	Yes	
INDEX, PRIME, OVERFLOW	PRIME or OVERFLOW (on different volume from INDEX)	No	Format 2 block built for INDEX during previous pass. Both the index area and the format 2 block will be on a different volume from the one currently being searched. If the same volume had been specified for both the INDEX request and the current request, a duplicate name format 1 block would have been encountered, and module IGG032I3 would have made an exit to module IGG032I4.
INDEX, PRIME	INDEX	Yes	
INDEX, PRIME	PRIME (on different volume from INDEX)	No	Format 2 block built for INDEX during previous pass. If the same volume had been specified, a format 1 block would have been encountered causing the subsequent exit as described above.
PRIME, OVERFLOW	PRIME	Yes	For this type of request, the first load module of the indexed sequential access method (ISAM) will construct an index in the overflow area.
PRIME, OVERFLOW	OVERFLOW	No	Format 2 block built for PRIME during previous pass. As previously described, a duplicate name format 1 block would have been encountered earlier in the processing by this module, and an exit would have been made.
PRIME	PRIME	Yes	The indexed sequential access method constructs the index area as described in the preceding entry above.

Note: If the DD card associated with a given indexed sequential data set has its 'type-of-space' parameter left blank, a request of PRIME is implied.

Figure 6. Conditions Affecting the Need for Module IGG032I3 to Build a Format 2 Block

After the extents have been created for the prime and the index areas, module IGG032I5 first adds up to three extents to the incompleted format 1 block. If more than three extents have been created, a format 3 block is built to contain all those beyond the first three. As extents are placed in the format 1 (and possibly the format 3) block, the index extents are entered first.

The DADSM table entries are converted from a relative track address (RTA) form to a cylinder and head (CCHH) form as they are entered in the DSCBs. A resident system convert routine is given control to do the conversion.

Since an index is being created, it is necessary to construct a format 2 block. If a format 3 block has been created, it is written out by a channel program initiated by module IGG032I5. Then the format 2 block is constructed, and the address of the format 3 block is placed in a pointer field. The format 2 block is then written out. If a format 3 block was not constructed, a format 2 block is constructed and written out after the last extent entry is placed in the format 1 block.

Once the format 2 block has been written out, its address is placed in a pointer field of the format 1 block, and the format 1 block is written out. As in modules IGG032I3 or IGG032I4, the channel program is set up to be used for reading in a format 5 DSCB, and control is given to module IGG032I6.

MODIFICATION OF A FORMAT 5 DSCB (MODULE IGG032I6): After the format 1 block, the format 2 block, and the format 3 block have been created and written out, the format 5 DSCB must be modified to reflect the changes in available space for other requests. Depending on the characteristics of the request, module IGG032I6 receives control from either module IGG032I3, module IGG032I4, or module IGG032I5. At the time this module is entered, main storage contains a DADSM table of up to five entries; each entry represents an extent that has been allocated for the given data set. The order of the entries both in the DADSM table and in a format 5 DSCB is the same as the order in which the corresponding areas are physically located on the direct access storage device. Each DADSM table entry is in the format of relative track beginning address and relative track ending address.

The identification of the allocated space indicated by the DADSM table entries must be removed from the corresponding format 5 DSCB. After removal of this space, the format 5 DSCB will represent only the space that is available for other requests.

The DADSM table entries may be related to format 5 DSCB extents in several ways:

1. A DADSM table entry may correspond exactly with a format 5 DSCB extent. In this case, when the format 5 DSCB is updated, the number of remaining extents is decreased by one.

2. A DADSM table entry and a format 5 DSCB extent may have the same beginning address, but the DADSM table entry's ending address is less than that of the format 5 DSCB extent. In this case, the number of remaining extents in the updated format 5 DSCB remains unchanged; only the extent's beginning address is changed.

3. A DADSM table entry and a format 5 DSCB extent may have the same ending address, but the DADSM table entry's beginning address is greater than that of the format 5 DSCB extent. In this case, the number of remaining extents in the updated format 5 DSCB remains unchanged; only the extent's ending address is changed.

4. A DADSM table entry may have a beginning address that is greater than that of a format 5 DSCB extent and an ending address that is less than that of the same format 5 DSCB extent. In this case, the updated format 5 DSCB will contain two new extents in place of the original one. One of the new extents will have the original beginning address but a different ending address. The other new extent will have the original ending address, but the beginning address will be different.

As each format 5 DSCB extent is removed, each of the following remaining extents is effectively "pushed up" nearer the beginning of the DSCB. If there are format 5 blocks chained together, the "push up" is extended across block boundaries and may result in the last format 5 block having no extents. The last format 5 block would then be deleted.

If format 5 DSCB extents are created, each of the following remaining extents in the DSCB (i.e., this includes both the block containing the created extent and all chained blocks if any exist) is "pushed down" in the block(s). This may result in the creation of a new format 5 block.

The section "Updating Format 5 Blocks" contains information about extent formats that is applicable in the case of the ISAM allocate routine.

After modifying the format 5 DSCB module IGG032I6 gives control to module IGG032I7.

COMPLETION OF PROCESSING (MODULE IGG032I7): The seventh load module (IGG032I7) of the ISAM allocate routine has several functions. It serves as the processing stage for errors encountered in the course of operation of any of the previous loads or within the seventh load itself. The format 4 DSCB for each given volume is updated to reflect the changes in the number of holes in the VTOC brought about by the creation of or deletion of DSCBs. When allocation has been completed and all

DSCBs have been built or updated, module IGG032I7 issues a DEQ macro instruction with the release-must-complete option for the VTOC. The seventh load module then returns control to the job scheduler.

If module IGG032I7 is entered as a result of a previous error, an indication is set to prohibit a later attempt by the control system to give control to the ISAM allocate routine. The main storage work areas assigned to the ISAM allocate routine are released and module IGG032I7 gives control to the job scheduler.

When the entrance to the seventh load module is by way of a normal (error-free) transfer from module IGG032I6, the format 4 DSCB that was read in by the first load of allocate is modified. The hole count of available blocks is updated, and the channel program writes out the updated format 4 DSCB.

If the current request is a single-volume request and it is the last of the up to three possible requests for a given data set allocation, the main storage work area is released and the TIOT fields are restored to their 'pre-ISAM allocate' condition, and control is given to the job scheduler. If the current request is not the last of a series of requests for a given data set allocation on a single volume, indicators are set to provide the necessary information for the subsequent requests, main storage areas are released and control returns to the job scheduler. The seventh load module processing of the last volume of a multi-volume request is the same as described for a single-volume request.

MULTI-VOLUME REQUESTS (MODULE IGG032I8): A multi-volume request requires the eighth load module to perform certain functions that parallel those performed by the first load module (IGC0003B) of the allocate routine. This is done to enable the ISAM allocate routine to maintain the counters and information which have been set up in main storage. The information relates to the multi-volume aspects of the request.

For the second volume (and for each succeeding volume in turn) module IGG032I8 issues a RESERVE macro instruction for the VTOC. A format 4 DSCB for the second volume (and for each succeeding volume, as required) is then read into main storage. A search is made for a duplicate name DSCB on this 'new' volume. If a valid duplicate name DSCB is found, control is given to the first module (IGG032I1) of the ISAM allocate routine. The processing then proceeds through the same steps as for a single volume request. A DADSM table is constructed for each volume, and each volume has its own format 5 DSCB which must be updated. An invalid duplicate name DSCB causes an exit to the error component of module IGG032I7.

THE EXTEND ROUTINE

If, when using the sequential access method to write records in a data set, the allocated space is exceeded, the extend routine dynamically allocates additional space to the data set. An indexed sequential data set cannot be extended. When a data set is extended, the extend routine may create a format 3 block and/or a format 1 block (if needed), take space from the format 5 DSCB, allocate that space to the data set, and update the VTOC. A data set may be extended up to five noncontiguous areas at a time. Extension of a data set that has 16 noncontiguous areas is not possible unless the data set can be extended to another volume.

If a sequential data set is extended onto another volume and user labels have been specified, the extend routine allocates an additional track on the new volume for the user label. This track will appear as the first extent described in the format 1 DSCB for this data set on the new volume. If the data set is Direct (DSORG=DA), a user label track is not allocated when the data set is extended onto another volume; the label appears only on the first volume of such a data set.

New blocks are created for an extended data set DSCB if:

1. The data set is extended to a new volume. After allocation, the VTOC on the new volume must contain at least a format 1 block for the data set.

2. The data set is extended to areas on the same volume, but the existing format 1 block does not contain enough space to hold the new extent entries. In this case, a format 3 block must be created.

To find space for a data set extension, the extend routine first tries to allocate space contiguous to the extent of the data set. If there is not enough contiguous space to fill the request, the routine builds an extent table of up to five noncontiguous areas that are smaller than, but closest in size to, the requested space. During the building of the extent table, the routine allocates the first area found that is equal to the request. If no area equals the request, the routine allocates the requested space from the area that is larger than, but closest in size to the request. If no areas are equal to or greater than the request, the routine allocates the request from as many of the five areas in the extent as necessary, beginning with the largest area. The request is always filled in the fewest possible number of noncontiguous areas.

EXTENDING SPLIT-CYLINDER DATA SETS

Split-cylinder data sets are extended differently from other data sets. The space for the extension is allocated on new cylinders, and only the extended data set can reside on those cylinders. The cylinders containing the extension are not regarded as part of the original split-cylinder extent, even though they may be contiguous to the original extent. No entry is made in the format 6 DSCB for split cylinder data set extensions.

THE EXTEND ROUTINE MODULES (MODULES IGG0553A, IGG0553B, IGG0553C, IGG0553D, AND IGG0553E)

The extend routine has five modules that are loaded separately into the transient area of main storage and operate in supervisor mode. The modules and the flow of control through them are shown in Figure 7. The functions performed by this routine are similar to those of the allocate routine; they consist of a duplicate name search, request conversion, extent allocation, DSCB creation, and format 5 DSCB updating. Input to the routine includes pointers to the DCB and DEB of the data set, the JFCB, and the serial number of the volume on which the data set is to be extended. The calling routine passes parameters to the extend routine by placing their addresses in general registers.

Module IGG0553A performs a duplicate name search to determine whether the data set already exists on the volume. The module issues a RESERVE macro instruction with the set-must-complete option for the VTOC, gets a work area from a protected subpool, and reads the format 4 DSCB from the VTOC. Then, the duplicate name search reads in the format 1 block of the data set's DSCB. If no format 1 block is found on the volume, a later module creates the block, which is then placed in the VTOC.

Except for record length requests the space required for the extension is obtained using the secondary quantity field in the JFCB (JFCBSQTY). For record length requests the JFCBBLKSI field is used. The input request is then converted into a track request. Record requests are rounded to the nearest track boundaries, and cylinder requests are aligned on cylinder boundaries.

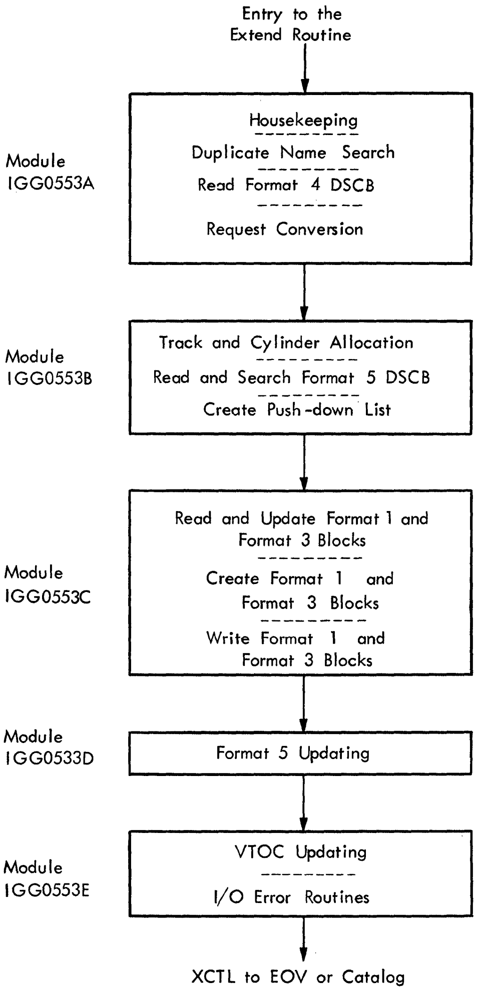

Figure 7. The Extend Routine Modules

Module IGG0553B reads format 5 blocks from the VTOC and tries to find space to fill the request. As each format 5 block is read, its entries are converted to relative track format to facilitate the allocation operation. The space allocated will be contiguous to the rest of the data set, if possible. Otherwise, it will be allocated, as in the allocate routine, in up to five noncontiguous areas that satisfy the request in the fewest areas. To satisfy a request for a user label, an additional track is obtained. This track must be contained within the five noncontiguous areas satisfying the request. These areas, or data set extents, are placed in an extent table known as the DADSM table. This table is passed to later modules in the extend routine.

Module IGG0553C updates and, if necessary, creates format 1 and format 3 blocks for the data set. Into these blocks are placed the selected area identifications that are obtained from the DADSM table. If a track was obtained for a user label, the first track of the first extent in the DADSM table becomes the first extent of the format 1 block for the data set. The format 1 and 3 blocks are then written in the VTOC. A channel program that is to be used in the next module is developed, and control passes to module IGG0553D.

Module IGG0553D initiates the channel program that was set up in the previous module, and updates the format 5 DSCB. The identification of the areas allocated to the data set is removed from the format 5 DSCB but remains in the extent table prepared in previous modules.

Module IGG0553E updates the format 4 DSCB, issues a DEQ macro instruction with the release-must-complete option for the VTOC, and exits to the calling routine. This module also contains common input/output error routines. When either module IGG0553B , module IGG0553C , or module IGG0553D encounter a permanent input/output error, control is transferred directly to this update module.

THE SCRATCH ROUTINE

The scratch routine reverses the procedure of initial allocation. If the data definition (DD) statement specifies deletion of a data set, the scratch routine will be entered at step termination time. The scratch routine will also be entered when the SCRATCH macro instruction is issued. Areas identified in the DSCB of the deleted data set are merged with the available areas accounted for in format 5 blocks. Format 0 blocks are written over the data set's DSCB.

Examples: Figure 8 refers to a hypothetical direct access device that has 40 tracks per cylinder. On this device, the last half of cylinder 5 and all of cylinder 6 together contain 4 extents and 7 areas as indicated. Data sets A, B, C, and E have one DSCB each in the VTOC of the volume. Each area is identified in one extent entry in the data set's DSCB. Available areas D and F are each identified by one extent entry in a format 5 DSCB in the VTOC. In each of the following examples, below, assume the initial condition shown in Figure 8.

1. If data set B were deleted, the extent from cylinder 5, track 30, to cylinder 6, track 10 would be accounted for in one extent entry in a format 5 DSCB. That entry would appear before the entries for areas D and F, to retain track address sequence in the format 5 DSCB. The DSCB for the data set would be deleted from the VTOC.

2. If data set C were deleted, the extent from cylinder 6, tracks 10 to 15, would be added to the available space from cylinder 6, tracks 15 to 25. The resulting area from cylinder 6, tracks 10 to 25, would be accounted for in one extent entry in a format 5 DSCB. That entry would appear before the entry for area F, to retain track address sequence in the format 5 DSCB. The DSCB for data set C would be deleted from the VTOC.

3. If data set E were deleted, the extent from cylinder 6, tracks 25 to 35, would be added to the area from cylinder 6, tracks 15 to 25, which in turn, would be added to the area from cylinder 6, tracks 35 to 40. The resulting area from cylinder 6, tracks 15 to 40, would be accounted for in one extent entry in a format 5 DSCB. The DSCB for data set E would be deleted from the VTOC.

```
                  Track
                  Addresses
                    20---┬--------------------┐
                         | Data Set A         |  One Area
                         |                    |  One Extent
         Cylinder 5 30---+--------------------+---------------
                         | Data Set B         |
                         |                    |
                    40---┴--------------------┘
                                                 One Extent
                                                 Two Areas
                     1---┬--------------------┐
                         | Data Set B         |
                         | Continued          |
                    10---+--------------------+---------------
                         | Data Set C         |  One Extent
                         |                    |  One Area
                    15---+--------------------+---------------
                         | Available Area D   |  One Area
                         |                    |
         Cylinder 6 25---+--------------------+---------------
                         | Data Set E         |  One Extent
                         |                    |  One Area
                    35---+--------------------+---------------
                         | Available Area F   |  One Area
                         |                    |
                    40---┴--------------------┴---------------
```

Figure 8. Data Sets on Two Cylinders

SCRATCHING OF SPLIT-CYLINDER DATA SETS

The extent allocated to split-cylinder data sets is not released for future allocations until all the data sets on the cylinders have been deleted. When one data set is scratched from a split-cylinder extent, the number 1 is subtracted from the fifth byte of the split-cylinder entry in the format 6 DSCB. When that fifth byte becomes zero, all data sets have been scratched from the split-cylinder extent, and the extent's areas are available for future allocation.

THE SCRATCH ROUTINE MODULES (MODULES IGC0002I, IGG0290A, IGG0290B, IGG0290C, IGG0290D AND IGG0290E)

The scratch routine consists of six modules, which are loaded and executed in the transient area of main storage in the sequence shown in Figure 9. The scratch routine modules use the XCTL table (see Appendix B) in the transferring of control from one module to the next.

When control is given to module IGC0002I at the beginning of the scratch routine, one of the general registers contains the address of a parameter list containing the following information:

- The address of the name of the data set to be deleted.

- The address of a list (the volume list) of volume serial numbers for the volumes (of the data set) to be deleted. See the "Note" in the discussion of the RENAME routine.

Module IGC0002I sequentially scans the volume serial numbers given in the unit control blocks (UCBs) that are indicated by a table of UCB addresses. These numbers are compared with the numbers that are referenced by the parameter list to determine if there is a UCB with a mounted volume that can be dismounted (e.g., one for which system considerations do not restrict dismounting). If there is, the address of the first such UCB encountered is saved for use by module IGG0290E, and control is given to module IGG0290E. If there is no volume that can be dismounted, control is given to module IGG0290E.

Module IGG0290E sequentially scans the volume serial numbers in the volume list, comparing each one with the volume serial numbers given in the UCBs. Each time a volume serial number match is found and the corresponding volume is already mounted, control is given to module IGG0290A (see Note 1).

If the volume corresponding to a serial number on the volume list is not mounted and module IGC0002I has indicated (by saving the UCB address) that a device is available for mounting, the volume is mounted on the device. Control is then given to module IGG0290A (see Note 1). If a device on which to mount a volume is not available, the portion of the data set on the volume cannot be deleted. The remaining entries in the volume list are examined in the same manner (i.e., to determine if they are already mounted and therefore permit the scratch routine to continue processing). If a volume cannot be mounted and there are no more entries in the volume list, control is returned to the calling routine.

Note: Prior to giving control to module IGG0290A to begin the actual deletion process, module IGG0290E issues an ENQ RESERVE macro instruction with the set-must-complete option for the VTOC of the volume.

Module IGG0290A searches the VTOC for the format 1 block of the data set. After finding a format 1 block, the module determines whether the data set is security protected. If it is protected, the module passes control to the system security checking routine (READPSWD), which checks the password and returns control. Module IGG0290A then determines whether the password was correct. If not, the module discontinues processing. Otherwise the module determines whether the data set's purge date is passed. If the purge-date condition is not satisfied, the module terminates processing. If the purge-date condition is satisfied, the module writes a format 0 block over the format 1 block. If the format 1 block has a chain address to another block of the DSCB, the module searches and destroys that block and any other block that is chained to it. Before each block is destroyed, the extent entries in that block are placed in an extent table that will be passed to the next module.

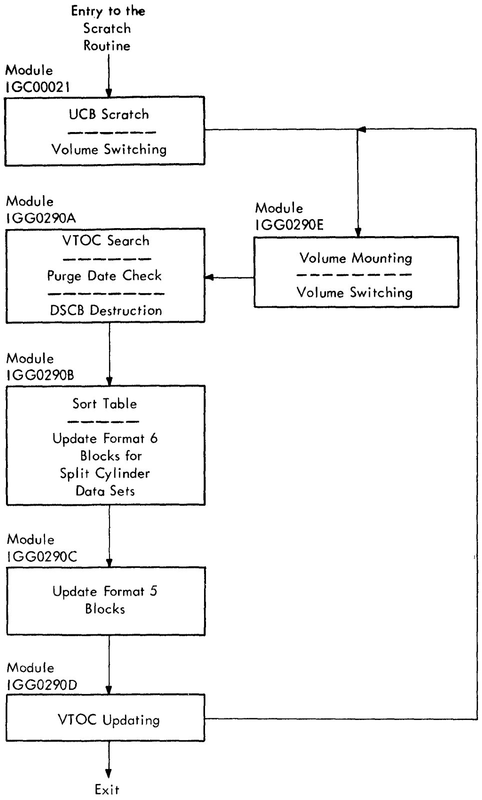

Figure 9. The Scratch Routine Modules

If the scratch routine can not scratch the data set from a volume for any reason, register 15 is loaded with an error code and the reason for the failure on each individual volume is indicated in the volume list that was input to the scratch routine. Control passes from module IGG0290A to module IGG0290B, where the entries in the extent table are sorted into ascending track and cylinder address sequence. If the data set is on split cylinders, module IGG0290B searches blocks of the format 6 DSCB in the VTOC for the appropriate split-cylinder extent entry. When the split-cylinder entry is found, the number 1 is subtracted from the total number of users on the split cylinders (contained in the fifth byte of the split-cylinder entry). If the result of the subtraction is zero, the split-cylinder entry is removed from the format 6 block and all subsequent split-cylinder entries must be moved to close the vacated space in that block. The area identification in the split-cylinder entry is placed in the extent table to be passed to the next module. If the split cylinders are not empty, the updated format 6 DSCB is written back into the VTOC with the split-cylinder entry remaining in the appropriate block. No area identification is placed in the extent table, because no area was made available.

Module IGG0290C merges the entries in the table with entries in format 5 blocks. The extent table is passed from the preceding module, and contains the identification of the areas vacated by the scratched data set. Format 5 blocks are then read, and the areas identified in the extent table are merged with those in the format 5 DSCB. As format 5 blocks are read, their extent entries are converted to the RTA1/RTA2 format, and before they are written back into the VTOC, they are converted back into the XXYYZ format. If the scratched data set is on split cylinders and no space was released, the merge is bypassed.

After processing the format 5 DSCB, module IGG0290C passes control to module IGG0290D which updates the VTOC's format 4 DSCB. During the scratch operation, a count is kept of the number of DSCB blocks deleted and created. If the number deleted does not equal the number created, the hole count in the format 4 DSCB must be changed. Module IGG0290D can then proceed to the next volume containing the data set, if there is another volume.

Before releasing control of the processing, module IGG0290D issues a DEQ macro instruction with the release-must-complete option for the VTOC of the current volume from which portions of the data set have been deleted. If there are no more volumes to process, all work areas are released and control returns to the calling routine. If there is another volume to be processed, work areas are re-initialized, and control passes to module IGG0290E.

THE RELEASE ROUTINE

The release routine deletes area identification from the DSCB of a data set, beginning at a specified track or cylinder and continuing to the end of the data set's extent. For example, suppose a data set was allocated three areas at the beginning of a job. During the job, a program wrote records up to the second track in the third area of the data set. If the job specified release of the unused space in the data set, the close routine of I/O support would request the release routine to release the tracks following the last used track of the third extent area. If either the original request was a cylinder request or the ROUND option was specified, the area released begins with the first cylinder following the last track that was used. The tracks released would be returned to available space for future allocation, as is done by the scratch routine.

When space is released from a data set, a format 3 block may be deleted from the VTOC. A format 3 block is deleted when all the extent areas identified in the block are returned as available space. A format 5 extent entry must be created to account for released space that is not contiguous with an available area.

The release routine cannot be specified for an indexed sequential data set. To do so would possibly result in the deletion of information in the overflow area.

If a DD statement specifies RELSE and the data set is opened for output and then closed with no data being written, the close routine of input/output support writes a file mark on the first track. In this case, the release routine releases all tracks (or cylinders) following the track containing the file mark.

The RELSE parameter applies whether the data set is old or new.

THE RELEASE ROUTINE MODULES (IGG020P1, IGG020P2, IGG020P3, AND IGG020D1)

The release routine contains four modules, which are loaded and executed in the transient area of main storage in the sequence shown in Figure 10. The routine is entered from the close routine of I/O support or from the reposition I/O routine of Checkpoint/Restart. The input consists of a format 1 block in the work area of the close routine, the DCB of the data set, and the identification of the UCB of the volume to be processed. Entry to the routine is in module IGG020P1, which gets a work area in a protected subpool. Before processing the VTOC, module IGG020P1 issues a RESERVE macro instruction with the set-must-complete option for the VTOC. If the format 1 block contains areas to be released, it is updated by module IGG020P1. Both the format 4 DSCB and the format 3 block (if present) are read into the work area. The identification of the released tracks is placed in an extent table to be passed to subsequent modules. Module IGG020P1 passes control to module IGG020P2.

Module IGG020P2 removes the identification of all released tracks from the format 3 block (if one is present). The format 3 block is then rewritten unless all extents have been removed from it. In the latter case, a format 0 block is written at the format 3 block location in the VTOC. The updated format 1 block is then written by the close routine. The first block of the format 5 DSCB is read into the work area, and control is passed to module IGG020D1.

Module IGG020D1 reads subsequent format 5 blocks and merges identification of released tracks from the extent table with area identification in the extent entries in format 5 blocks. It also updates the format 4 DSCB. Module IGG020P3 determines if the release routine was entered from CLOSE or Checkpoint/Restart and returns to the modules IGG0200F or IGC0005B, respectively. These two modules free the VTOC by issuing a DEQ macro instruction with the release-must-complete option for the VTOC.

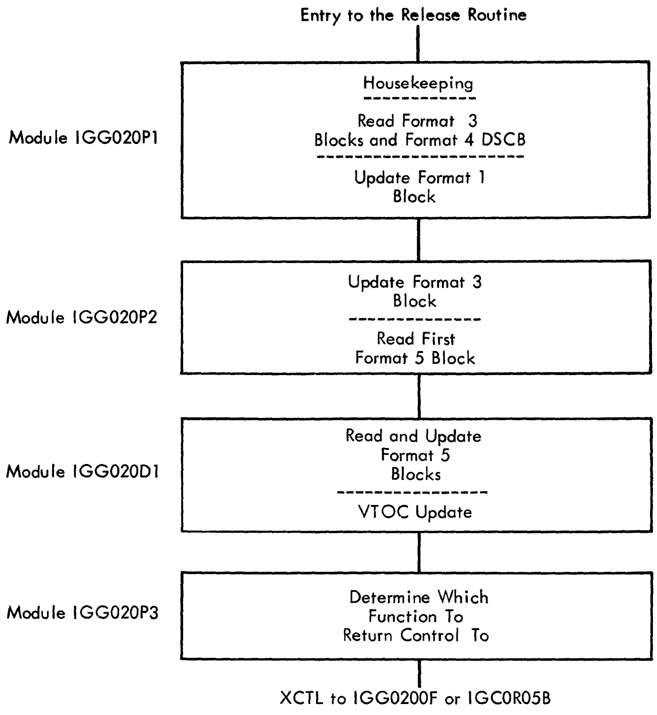
Entry to the Release Routine
Module IGG020P1
Housekeeping
Read Format 3 Blocks and Format 4 DSCB
Update Format 1 Block
Module IGG020P2
Update Format 3 Block
Read First Format 5 Block
Module IGG020D1
Read and Update Format 5 Blocks
VTOC Update
Module IGG020P3
Determine Which Function To Return Control To
XCTL to IGG0200F or IGC0R05B

Figure 10. The Release Routine Modules

THE RENAME ROUTINE

The rename routine consists of three modules, as shown in Figure 11. The routine is called by the RENAME macro instruction. Input to the first module, IGC00030, includes a general register that contains the address of a parameter list containing the following information:

- The old (current) name of the data set to be renamed.
- The new name of the data set to be renamed.
- The address of a list (the volume list) of volume serial numbers for the volumes on which the data set that is to be renamed resides. Note: Each of the DADSM routines that can modify the VTOC contains a data extent block (DEB) in a work area. The DEB contains the address of the UCB for the volume currently being modified, and the UCB contains the serial number of the volume.

In order for the rename routine to process a multivolume data set, either all of its volumes must be mounted, or else one mounted volume must be demountable, physically and logically.

Module IGC00030 sequentially scans the volume serial numbers given in the unit control blocks that are indicated by a table of UCB addresses. The serial numbers are compared with the numbers that are referenced by the parameter list to determine if there is a UCB with a mounted volume (for renaming) that can be dismounted (as far as system considerations are concerned). If there is, the address of the first such UCB encountered is saved for use by module IGG03002, and control is given to module IGG03002. If there is no volume available for demounting, control is given to module IGG03002.

Module IGG03002 sequentially scans the volume serial numbers in the volume list, comparing each one with the volume serial numbers given in the UCBs. Each time a volume serial number match is found and the corresponding volume is already mounted, control is given to module IGG03001.

If the volume corresponding to a serial number on the volume list is not mounted and module IGC00030 has indicated (by saving the UCB address) that a device is available for mounting, the volume is mounted on the device. Control is then given to module IGG03001. If a device on which to mount a volume is not available, the portion of the data set on the volume cannot be renamed. The remaining entries on the volume list are examined in the same manner (i.e., to determine if they are already mounted and therefore permit the rename routine to continue processing). If a volume cannot be mounted and there are no more entries on the volume list, control is returned to the calling routine.

Module IGG03001 reads the format 1 block of the data set's DSCB. (Note: When operating in a shared DASD environment, module IGG03001 first issues a RESERVE option for the VTOC of the volume that contains the data set to be renamed. This procedure protects the VTOC from modification by other tasks until the rename function is completed for that volume.) The new name is placed in the block, and the block is placed into the VTOC at the same place from which it was read.

Module IGG03001 then issues a DEQ macro instruction (with the release-must-complete option) for the VTOC of the volume being modified. If the operating system environment is using the shared direct access storage device (shared DASD) option, the macro instruction also results in the release of the device containing the volume whose VTOC is being modified.

If the data set is contained on more than one volume, and all volumes have not been processed, control returns to module IGG03002. Otherwise (after releasing the VTOC of the volume by using the DEQ macro instruction), control passes back to the calling routine.

If the data set to be renamed is security protected, module IGG03001 transfers control to the security checking routine (READPSWD). READPSWD indicates whether the password is valid. If the password is valid, module IGG03001 completes the rename function as previously described. If the password is invalid, module IGG03001 indicates the error and returns control to the calling routine as before.

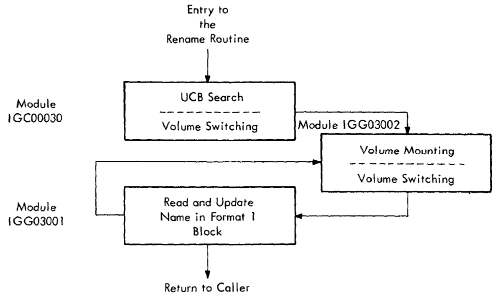

Figure 11. The Rename Routine Modules

THE OBTAIN ROUTINE

The obtain routine is initiated by the execution of the OBTAIN macro instruction. Input to the routine, passed by general register settings, consists of the address of a parameter list that contains the serial number of the volume to be processed, an indication of either the seek or search option, and either the name of a data set or the address of a block in the VTOC. The routine must also be provided with a work area.

The routine consists of one module. If a fully qualified name is provided, this module, IGC0002G, searches the VTOC and reads in the format 1 block that contains that name. The name option will result in the reading of the data portion of the named format 1 block and the address of the block. If an absolute track address is provided in the input, the module constructs and initiates a channel program to read the key and data portions of the block directly from that address. When supplied with an absolute track address, the obtain routine will not process the DSCB unless it is in the boundaries of the extent of the VTOC.

The Lspace routine consists of one module, IGC0007H, and is entered directly through an SVC 78. The routine expects one general register to point to a UCB (sub-UCB in the case of a 2321), and another to point to a 30-byte area into which it will put the requested space information.

The routine checks first for a valid UCB address, checks the UCB to ensure that it identifies a direct access device, and checks the message area address to be sure it is in the user's area. It then obtains a work area in which it builds channel programs and their associated control blocks. It reserves the VTOC to guarantee that the information returned to the user is meaningful.

The routine then reads the format 4 DSCB and the first format 5. It checks the format 4 to be sure that the volume being examined is a standard OS volume and then begins calculating available space. It does this by going through the extent descriptions in each format 5 DSCB and totaling the free space described. The routine always maintains a record of the largest contiguous extent encountered and updates it when a larger extent is encountered.

When the last format 5 DSCB has been checked, the routine returns a code of zero in register 15, assembles the space information in its final format (see below), frees core, DEQs on the VTOC, and returns to the caller.

If no errors have occurred, the user's area will contain a message in EBCDIC in the following format:

```
SPACE=CCCC,TTTT,AAAA/cccc,tttt
```

Where:

CCCC	=	Total number of free cylinders.
TTTT	=	Total number of free tracks.
AAAA	=	Total number of free extents.
cccc, tttt	=	Largest contiguous free area in cylinders and tracks.

If an error has occurred, an error code will be placed in register 15, and an error message may be placed in the user's message area. See Figure 12 for a description of the error messages.

Code	Message	Condition
0	See above	Normal processing
4	LSPACE-PERMANENT I/O ERROR	I/O error while reading DSCBs
8	LSPACE-NON-STANDARD OS VOLUME	Volume's last use was not under OS/360
12	LSPACE-INVALID PARAMETER (or) LSPACE-NOT A DIRECT ACCESS VOL	UCB address invalid UCB not for direct access device
16	None	Invalid message area address

Figure 12. Lspace Error Messages

APPENDIX A: THE EXTENT TABLE FORMAT

This appendix describes the extent (or, DADSM) table used to pass information between modules of the allocate, extend, scratch, and release routines. Figure 13 shows the format of the extent table. The extent table used in the allocate and extend routines is 24 bytes long and can contain up to 5 extent entries. The table used in the scratch and release routines is 68 bytes long to accommodate up to 16 extent entries.

The information contained in the flags field differs for each routine and can be determined by examining the program listing.

The number of entries field contains the count of the number of extent entries in the extent table.

The used hole field contains the count of the number of new blocks used to record any DSCBs in the VTOC during allocation or scratching.

Each extent entry contains two binary numbers: RTA1 and RTA2. RTA1 is the relative track address of the first track of an extent area: RTA2 is the relative track address of the last track of the area, plus one. A relative track address is a binary number representing the displacement of the track from the first track of the volume. For example, the relative track address of track seven, cylinder zero is seven. The relative track address of track 3 on cylinder 2 is 23, if the volume has ten tracks per cylinder.

FLAGS	Number of Entries	Used Hole Count	RTA1	RTA2	Remaining Portion Contains up to 15 Extent Entries of One Word Each

```
0       1       2     3 4             5 6             7 8
                        |<-------One Extent Entry------>|
```

Figure 13. The Extent Table Format

The XCTL table is used in the transferring of control: (1) between modules of the allocate routine for non indexed-sequential data sets, and (2) between modules of the scratch routine. The format of the XCTL table is shown in Figure 14. There is an XCTL table starting on a double-word boundary in each load. The table consists of the other load IDs to which a given load can transfer control. Each entry consists of the load ID, its relative disk address (TTR) and the length of the load. The TTRs are inserted by the IEHIOSUP utility program when the system is generated. The last four bytes of a load consist of a supervisor-call (SVC) pointing to the beginning of the XCTL table. The pointer is expressed in double words from the beginning of the load.

Load	Program	
Load ID$_1$ (2 bytes)	Relative disk address (TTR) (3 bytes)	Length of load expressed in double words (1 byte)
Load ID$_2$	TTR of load	Length
Load ID$_3$	TTR of load	Length
Load IDs continue		
Load ID$_n$	TTR of load	Length
00 (end of table)		
	(Program continues)	
	SVC code	Pointer to the first ID$_1$ relative to 0 and expressed in double words
	|<-------------------- last 4 bytes of load -------------------->|	

Figure 14. XCTL Table

APPENDIX C: SUMMARY OF DADSM ROUTINES

Figure 15 contains the names of the DADSM routines, the modules included within each routine, the primary function of each routine, and the method by which each routine is invoked.

Name of Routine	Modules in the Routine	Function of the Routine	Method of Invoking the Routine
Allocate	IGC0003B IGG0325B IGG0325C IGG0325D IGG0325E IGG0325F IGG0325G IGG0325H IGG0325J IGG0325K IGG0325L IGG0325S	Allocates initial space to data sets.	Use of SVC 32, which is issued only by the task scheduler or by utility programs.
[Allocate] (DOS to OS VTOC Conversion)	IGG0325Z IGG0325R IGG0325P IGG0325Q IGG0325T	Converts DOS VTOC to OS format.	
(Indexed Sequential)	IGG032I2 IGG032I3 IGG032I4 IGG032I5 IGG032I6 IGG032I7 IGG032I8		
Extend	IGG0553A IGG0553B IGG0553C IGG0553D IGG0553E	Allocates additional space to data sets when an end-of-volume condition has been reached. (Note: Indexed sequential data sets cannot be extended.)	Use of the XCTL macro-instruction by either the catalog or the end-of-volume routines.
Scratch	IGC0002I IGG0290A IGG0290B IGG0290C IGG0290D IGG0290E	Deletes data sets from volumes by removing their associated DSCBs from VTOCs.	Use of SVC 29, which is issued by either the task scheduler, directory management, or utility programs.
Release	IGG020P1 IGG020P2 IGG020P3 IGG020D1	Returns unused data set space to available space (accounted for by format 5 DSCBs) when a data set is closed.	Use of XCTL macro-instruction by the close routine of I/O support.
Rename	IGC00030 IGG03002 IGG03001	Changes the name of a data set.	Use of SVC 30.
Obtain	IGC0002G	Provides access to any DSCB in the VTOC.	Use of SVC 27.
Lspace	IGC0007H	Provides user with information about available space.	Use of SVC 78.

●Figure 15. The DADSM Routines

INDEX

Where more than one page reference is given, the major reference is first.

READER'S COMMENT FORM

IBM System/360 Operating System
Direct Access Device Space Management
Program Logic Manual

Form Y28-6607-5

- Your comments, accompanied by answers to the following questions, help us produce better publications for your use. If your answer to a question is "No" or requires qualification, please explain in the space provided below. Comments and suggestions become the property of IBM.

	Yes	No
• Does this publication meet your needs?	□	□
• Did you find the material:		
Easy to read and understand?	□	□
Organized for convenient use?	□	□
Complete?	□	□
Well illustrated?	□	□
Written for your technical level?	□	□

- What is your occupation? ____________________
- How do you use this publication?

As an introduction to the subject?	□	As an instructor in a class?	□
For advanced knowledge of the subject?	□	As a student in a class?	□
For information about operating procedures?	□	As a reference manual?	□

Other ____________________

- Please give specific page and line references with your comments when appropriate.

COMMENTS

- Thank you for your cooperation. No postage necessary if mailed in the U.S.A.

YOUR COMMENTS, PLEASE...

This publication is one of a series which serves as reference sources for systems analysts, programmers and operators of IBM systems. Your answers to the questions on the back of this form together with your comments, will help us produce better publications for your use. Each reply will be carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become the property of IBM.

Please note: Requests for copies of publications and for assistance in utilizing your IBM system should be directed to your IBM representative or to the IBM sales office serving your locality.

fold fold

FIRST CLASS
PERMIT NO. 2078
SAN JOSE, CALIF.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN U. S. A.

POSTAGE WILL BE PAID BY . . .

IBM Corporation
Monterey & Cottle Rds.
San Jose, California
95114

Attention: Programming Publications, Dept. D78

fold fold

IBM®

International Business Machines Corporation
Data Processing Division
112 East Post Road, White Plains, N.Y. 10601
[USA Only]

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
[International]

IBM S/360 Printed in U.S.A. Y28-6607-5

IBM®

International Business Machines Corporation
Data Processing Division
112 East Post Road, White Plains, N.Y. 10601
[USA Only]

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
[International]